

LDPC-Based Iterative Algorithm for Compression of Correlated Sources at Rates Approaching the Slepian-Wolf Bound

Fred Daneshgaran, Massimiliano Laddomada, and Marina Mondin,

Abstract

This article proposes a novel iterative algorithm based on Low Density Parity Check (LDPC) codes for compression of correlated sources at rates approaching the Slepian-Wolf bound. The setup considered in the article looks at the problem of compressing one source at a rate determined based on the knowledge of the mean source correlation at the encoder, and employing the other correlated source as side information at the decoder which decompresses the first source based on the estimates of the actual correlation. We demonstrate that depending on the extent of the actual source correlation estimated through an iterative paradigm, significant compression can be obtained relative to the case the decoder does not use the implicit knowledge of the existence of correlation.

Index Terms

Correlated sources, compression, iterative decoding, joint decoding, low density parity check codes, Slepian-Wolf, soft decoding.

I. INTRODUCTION

Consider two independent identically distributed (i.i.d.) discrete binary memoryless sequences of length k , $X = [x_1, x_2, \dots, x_k]$ and $Y = [y_1, y_2, \dots, y_k]$, where pairs of components (x_i, y_i) have joint probability mass function $p(x, y)$. Assume that the two sequences are generated by two transmitters which do not communicate with each other, and that both sequences have to be jointly decoded at a common receiver. Slepian and Wolf [1] demonstrated that the achievable rate region for this problem (i.e., for perfect recovery of both sequences at a joint decoder), is the one identified by the following set of equations imposing constraints on the rates R_X and R_Y at which both correlated sequences are transmitted:

$$\begin{cases} R_X \geq H(X|Y), \\ R_Y \geq H(Y|X), \\ R_X + R_Y \geq H(X, Y) \end{cases} \quad (1)$$

whereby $H(X|Y)$ is the conditional entropy of source X given source Y , $H(Y|X)$ is the conditional entropy of source Y given source X , and $H(X, Y)$ is the joint entropy. A pictorial representation of this achievable region is given in Fig. 1-a.

This work has been supported by Euroconcepts S.r.l. (<http://www.euroconcepts.it>)

F. Daneshgaran is with the ECE Dept., Calif. State Univ., Los Angeles, USA.

M. Laddomada and M. Mondin are with the Dipartimento di Elettronica, Politecnico di Torino, Italy.

In this article, we focus on trying to achieve the corner points A and B in Fig. 1-a, since any other point between these can be achieved with a time-sharing approach [1]. In particular, we focus on the architecture shown in Fig. 1-b in which we assume that one of the two sequences, namely X in our framework, is independently encoded with a source encoder that has the knowledge of the mean correlation between the sources X and Y . We assume that sequence Y is compressed up to its source entropy $H(Y)$ and is known at the joint decoder as side information, and our aim is at compressing sequence X with a rate R_X as close as possible to its conditional entropy $R_X \geq H(X|Y)$ in order to achieve the corner point A in Fig. 1-a. The decoder tries to decompress the sequence X , in order to obtain an estimate \hat{X} , by employing Y as side information. As shall be seen shortly, the decoder has an implicit knowledge of mean correlation between sources from the block length of the encoded sequence. It estimates the actual correlation between the two sequences through an iterative algorithm which improves the decoding reliability of X . Obviously, our solution to joint source coding at point A is directly applicable to point B by symmetry. The overall rate of transmission of both sequences is greater than $H(Y) + H(X|Y) = H(X, Y)$.

With this background, let us provide a quick survey of the recent literature related to the problem addressed in this article. This survey is by no means exhaustive and is meant to simply provide a sampling of the literature in this area.

In [2], the authors show that turbo codes can allow one to come close to the Slepian-Wolf bound in lossless distributed source coding. In [3], [4], the authors propose a practical coding scheme for separate encoding of the correlated sources for the Slepian-Wolf problem. In [5], the authors propose the use of punctured turbo codes for compression of correlated binary sources whereby compression has been achieved via puncturing. The proposed source decoder utilizes an iterative scheme to estimate the correlation between two different sources. In [6], punctured turbo codes have been applied to the compression of non-binary sources.

Paper [7] deals with the use of parallel and serial concatenated convolutional codes as source-channel codes for the transmission of a memoryless binary sequence with side information at the decoder, while in [8], [9] the authors propose a practical coding scheme based on LDPC codes for separate encoding of the correlated sources for the Slepian-Wolf problem. The problem of Slepian-Wolf correlated source coding over noisy channels has been dealt with in papers [10]-[14].

Relative to the cited articles, the main novelty of the present work may be summarized as follows: 1) in [5] and [9] the encoder and decoder must both know the correlation between the two sources. We assume knowledge of mean correlation at the encoder. The decoder has implicit knowledge of this via observation of the length of the encoded message. It iteratively estimates the *actual* correlation observed and uses it during decoding; 2) our algorithm can be used with any pair of systematic encoder/decoder without modifying the encoding and decoding algorithm; 3) the proposed algorithm is very efficient in terms of the required number of LDPC decoding iterations. We use quantized integer LLR values (LLRQ) and the loss of our algorithm for using integer LLRQ metrics is quite negligible in light of the fact that it is able to guarantee performance better than that reported in [5] and [9] (where, to the best of our knowledge, authors use floating point metrics) as exemplified by the results shown in table II below; 4) we utilize post detection correlation estimates to generate extrinsic information, which can be applied to any already employed decoder without any modification; and 5) we do not use any interleaver between the sources at the transmitter. Using the approach of [5] in a network, information about interleavers used by different nodes must be communicated and managed. This is not trivial in a distributed network such as the internet. Furthermore, there is a penalty in terms of delay that is incurred.

This section focuses on the source encoder used for source compression. LDPC coding is essential to achieving performance close to the theoretical limit in [1]. The LDPC matrix [15] for encoding each source is considered as a systematic (n, k) code. The codes used need to be systematic for the decoder to exploit the estimated correlation between X and Y directly. Each codeword C is composed of a systematic part X , and a parity part Z which together form $C = [X, Z]$. With this setup and given the parity check matrix $H^{n-k, n}$ of the LDPC code, it is possible to decompose $H^{n-k, n}$ as follows:

$$H^{n-k, n} = (H^X, H^Z) \quad (2)$$

whereby H^X is a $(n-k) \times (k)$ matrix specifying the source bits participating in check equations, and H^Z is a $(n-k) \times (n-k)$ matrix of the form:

$$H^Z = \begin{pmatrix} 1 & 0 & \dots & 0 & 0 \\ 1 & 1 & 0 & \dots & 0 \\ 0 & 1 & 1 & 0 & \dots \\ \dots & \dots & \dots & \dots & \dots \\ 0 & \dots & 0 & 1 & 1 \end{pmatrix}. \quad (3)$$

The choice of this structure for H , also called staircase LDPC (for the double diagonal of ones in H^Z), has been motivated by the fact that aside from being systematic, we obtain a LDPC code which is encodable in linear time in the codeword length n . In particular, with this structure, the encoding operation is as follows:

$$z_i = \begin{cases} \left[\sum_{j=1}^k x_j \cdot H_{i,j}^X \right] \pmod{2}, & i = 1 \\ \left[z_{i-1} + \sum_{j=1}^k x_j \cdot H_{i,j}^X \right] \pmod{2}, & i = 2, \dots, n-k \end{cases} \quad (4)$$

where $H_{i,j}^X$ represents the element (i, j) of the matrix H^X , and x_j is the j -th bit of the source sequence X .

Source compression is performed as follows; considering the scheme shown in Fig. 1-b, we encode the length k source sequence X and transmit on a perfect channel only the parity sequence Z , whose bits are evaluated as in (4). The rate guaranteed by such an encoder is $R_X = \frac{n-k}{k}$. In relation to the setup shown in Fig. 1-b, the Slepian-Wolf problem reduces to that of encoding the source X with a rate R_X as close to $H(X|Y)$ as possible (i.e., $R_X \geq H(X|Y)$). The objective of the joint decoder is to recover sequence X by employing the correlated source Y (considered as side information at the decoder), and the estimates of the actual correlation between the sources X and Y obtained in an iterative fashion.

We consider the following model in order to follow the same framework pursued in the literature [5], [8]:

$$P(x_j \neq y_j) = p, \quad \forall j = 1, \dots, k \quad (5)$$

In light of the considered correlation model, and noting that the sequence Y is available losslessly at the joint decoder ($R_Y = 1$), the theoretical limit for lossless compression of X is $R_X \geq H(X|Y) = H(p)$, whereby $H(p)$ is the binary entropy function.

Note that the encoder needs to know the mean correlation so as to choose a rate close to $H(p)$. It does so, by keeping k constant while choosing n appropriately. We use the term mean correlation, because in any actual setting, the exact correlation between the sequences may be varying about the mean value. Hence, it is beneficial if the decoder estimates the actual correlation value from observations itself. While no side information about the rate is communicated to the decoder, the decoder knows the mean correlation implicitly from the knowledge of block length n .

The architecture of the iterative joint decoder for the Slepian-Wolf problem is depicted in Fig. 1-c. Its goal is to determine the best estimate \hat{X} of the source k -sequence X , by starting from the received parity bit sequence Z of length $(n - k)$.

Based on the notation above, we can now develop the algorithm for exploiting the source correlation in the LDPC decoder. Consider a (n, k) -LDPC identified by the matrix $H^{(n-k, n)}$ as expressed in (2). Note that we only make reference to maximum rank matrix H since the particular structure assumed for H ensures this. In particular, the double diagonal on the parity side of the H matrix always guarantees that the rank of H is equal to the number of its rows, i.e., $n - k$.

For conciseness, we will present only the modifications to the classical belief-propagation algorithm. The main modification concerns the initialization step whereby in our setup, each bit-node is assigned an a-posteriori LLR as follows:

$$L(u_j) = \begin{cases} \log\left(\frac{P(x_j=1|y_j)}{P(x_j=0|y_j)}\right) = (2y_j - 1)\alpha^{(i)}, & j = 1, \dots, k \\ (2z_j - 1), & j = k + 1, \dots, n \end{cases} \quad (6)$$

where $\alpha^{(i)} = \log\left(\frac{p^{(i)}}{1-p^{(i)}}\right)$ is the correction factor taking into account the estimated correlation between sequences X and Y at global iteration i . Note that this term derives from the correlation model adopted in this paper as expressed in (5), in which the correlation between any bit in the same position in the two sequences X and Y is seen as having been produced by an equivalent binary symmetric channel with transition probability p .

The architecture of the iterative joint decoder is depicted in Fig. 1-c. We note that there are two stages of iterative decoding. Index i denotes a *global iteration* whereby during each global iteration, the updated estimate of the actual source correlation obtained during the previous global iteration is passed on to the belief-propagation decoder that performs *local iterations* with a pre-defined stopping criterion and/or a maximum number of local decoding iterations.

Let us elaborate on the signal processing involved. In particular, as before let x and y be two correlated binary random variables which can take on the values $\{0, 1\}$ and let $r = x \oplus y$. Let us assume that random variable r takes on the values $\{0, 1\}$ with probabilities $P(r = 1) = p_r$ and $P(r = 0) = 1 - p_r$.

The correction factor $\alpha^{(i)}$ at global iteration (i) is evaluated as follows,

$$\alpha^{(i)} = \log\left(\frac{p_{\hat{r}}}{1 - p_{\hat{r}}}\right), \quad (7)$$

by counting the number of places in which $\hat{X}^{(i)}$ and Y differ, or equivalently by evaluating the Hamming weight $w_H(\cdot)$ of the sequence $\hat{R}^{(i)} = \hat{X}^{(i)} \oplus Y$ whereby, in the previous equation, $p_{\hat{r}} = \frac{w_H(\hat{R}^{(i)})}{k}$. In the latter case, by assuming that the sequence $\hat{R} = \hat{X} \oplus Y$ is i.i.d., we have:

$$\alpha^{(i)} = \log\left(\frac{w_H(\hat{R}^{(i)})}{k - w_H(\hat{R}^{(i)})}\right) \quad (8)$$

where k is the source block size. Above, letters highlighted with $\hat{\cdot}$ are used to mean that the respective parameters have been estimated.

Formally, the iterative decoding algorithm can be stated as follows:

- 1) Set the log-likelihood ratios $\alpha^{(0)}$ to proper initial values based on the knowledge of the mean source correlation (see Fig. 1-c). Compute the log-likelihood ratios for any bit node using (6).
- 2) For each global iteration $i = 1, \dots, M$, do the following:

- a) perform belief-propagation decoding on the parity bit sequence Z by using a predefined maximum number of local iterations, and the side information represented by the correlated sequence Y along with the correction factor $\alpha^{(i-1)}$;
- b) Evaluate $\alpha^{(i)}$ using (8);
- c) If $|\alpha^{(i)} - \alpha^{(i-1)}| \geq 10^{-4}$ go back to (a) and continue iterating, else exit.

Step c) in the previous code fragment is used in order to speed-up the overall iterative algorithm. Extensive tests we conducted suggested that the threshold value of 10^{-4} may be used for this purpose. Obviously, one can keep iterating until the last global iteration as well.

A. Overview of Integer-Metrics Belief-Propagation Decoder

In this section, we briefly describe the LDPC decoder working with integer LLRs. This approach leads to efficient belief-propagation decoding. We begin by quantizing any real LLR (denoted LLRQ after quantization) employed in the initialization phase of the belief-propagation decoder in (6), using the following transformation:

$$LLRQ = \begin{cases} \lfloor 2^q L(u_j) + 0.5 \rfloor, & j = 1, \dots, k \\ \lfloor 2z_j - 1 \rfloor \cdot S, & j = k + 1, \dots, n \end{cases} \quad (9)$$

whereby $\lfloor \cdot \rfloor$ stands for rounding to the smaller integer in the unit interval in which the real number falls, $L(u_j)$ is the real LLR, S is a suitable scaling factor, and q is the precision chosen to represent the LLR with integer metrics. In our belief-propagation decoder, we use $q = 3$, which guarantees a good trade-off between BER performance and complexity of the decoder implementation. The scaling factor S is the greatest integer metric processed by the iterative decoder. In our set-up, we use $S = 10000$. Note that such a scaling factor depends on the practical implementations of the belief-propagation decoder. Suffice it to say that in our setup, S gives high likelihood to the parity bits z_j , $\forall j = k + 1, \dots, n$, since they are transmitted through a perfect channel to the decoder.

IV. SIMULATION RESULTS AND COMPARISONS

We have simulated the performance of our proposed iterative joint source decoder. We follow the same framework as in [5], [8], [9].

In the following, we provide sample simulation results associated with various (n, k) LDPC codes designed with the technique proposed in [16]. In particular, for a fair comparison with the results provided in [9], we designed various LDPC codes with source block length $k = 16400$. The details and the parameters of the designed LDPCs are given in Table I.

Parameters given in Table I are the source block length k , the codeword length n , the rate R_X of the source, expressed as $\frac{n-k}{k}$ (i.e., inverse of the compression ratio), the average degree d_v of the bit nodes, and the average degree d_c of the check nodes of the designed LDPCs. Note that, the encoding procedure adopted in our approach is different from the one proposed in [9] in that we source encode k bits at a time and transmit only $n - k$ bits. In [9], the authors proposed a source compression which encodes n source bits at a time, and transmits $n - k$ syndrome bits.

For local decoding of the LDPC codes, the maximum number of local iterations has been set to 50, while the maximum number of global iterations is 5, even though the stopping criterion discussed in the previous section has been adopted.

In order to test the proposed algorithm for varying actual correlation levels, for any given value of mean correlation p , we generate a uniform random variable having mean value equal to the mean correlation itself and with a maximum variation of

Δp around this mean value. We used the following maximum variations: $\Delta p = 0.5, 0.2, 0.1\%$, and $\Delta p = 0.0\%$ which refers to the case in which the correlation value is not variable, but fixed.

For each data block, we set the actual correlation equal to the mean correlation plus this perturbation. The decoder iterates to estimate the actual correlation value which varies around its mean value from one block to the next. In effect, the parameter p is iteratively estimated as discussed in the previous section. A similar approach has been pursued in [5] for fixed correlation level, whereby an iterative approach is used for the estimation of the correlation between the two correlated sequences, but employing turbo codes.

Finally, note that we employ integer soft-metrics as explained in the previous section, while in [5], [9], to the best of our knowledge, the authors employ real metrics. The algorithm working on integer metrics is very fast and reduces considerably the complexity burden required by the two-stage iterative algorithm (i.e., the local-global combination).

Fig. 2 shows the BER performance of the proposed iterative decoding algorithm for a maximum of 5 global iterations and as a function of the joint entropy between sources X and Y , when the stopping criterion for global iterations is applied. LDPCs used for encoding are the one labelled L_3 and L_4 in Table I which guarantee compression rates of $R_X = 0.237$ and $R_X = 0.189$, respectively. LDPC labelled L_3 is used at mean values of p equal to 0.025, while LDPC L_4 is adopted for a mean correlation of 0.015. From Fig. 2 one clearly sees that LDPC decoding does not converge when the decoder does not iterate for estimating the actual value of p , but uses only its mean value for setting the extrinsic information. Notice also that the performances of the iterative decoder when the correlation value is fixed (curves labelled $\Delta p = 0.0$ in Fig. 2), are very close to the case in which the actual correlation value varies within $\Delta p = 0.1\%$ from the mean value.

Similar considerations can be deduced from Fig. 3 which shows the BER performance of the proposed iterative decoding algorithm when using LDPCs labelled L_1 and L_2 in Table I which guarantee compression rates of $R_X = 0.597$ and $R_X = 0.365$, respectively. LDPC labelled L_1 is used at mean correlation equal to 0.1, while LDPC L_2 is used with a mean correlation of 0.05. Note that the performance degrades as Δp increases since the encoder works further away from its optimal operating point.

Finally, we evaluated the average number of global iterations performed by the iterative algorithm when the stopping criterion on global iterations is employed during decoding. Simulation results show that when the LDPC decoder works at BER levels below 10^{-5} , the average number of global iterations equals 1.2, thus guaranteeing a very efficient iterative approach to the co-decompression problem. In other words, an overall average number of 80 LDPC decoding iterations suffices to obtain good BER performance.

The results on the compression achieved with the proposed algorithm are shown in Table II for the case in which the correlation value is fixed. The first row shows the fixed correlation parameter assumed, namely, $p = P(x_j \neq y_j), \forall j = 1, \dots, k$ in our model. The second row shows the joint entropy limit for various values of the fixed correlation parameter p . The third and fourth rows show the results on source compression presented in papers [5], [9], while the last row presents the results on compression achieved with the proposed algorithm employing a maximum of 5 global iterations in conjunction with using the stopping criterion noted in the previous section. As in [9], we assume error free compression for a target Bit Error Rate (BER) 10^{-6} . Note that statistic of the results shown has been obtained by counting 30 erroneous frames.

From Table II it is evident that significant compression gains with respect to the theoretical limits can be achieved as the

correlation between sequences X and Y increases.

V. CONCLUSIONS

In this article we have presented a novel iterative joint decoding algorithm based on LDPC codes for the Slepian-Wolf problem of compression of correlated information sources. In the considered scenario, two correlated sources communicate with a common receiver. The first source is compressed by transmitting the parity check bits of a systematic LDPC encoded codeword. The correlated information of the second source is employed as side information at the receiver and used for decompressing and decoding of the first source. The crucial observation is that LDPC decoding does not converge when the decoder does not iterate for estimating the actual value of p , but uses instead its mean value which is assumed to be implicitly known. Both the iterative decoding algorithm and the cross-correlation estimation procedure have been described in detail. Simulation results suggest that relatively large compression gains are achievable at relatively small number of global iterations specially when the sources are highly correlated.

REFERENCES

- [1] D. Slepian and J. Wolf, "Noiseless coding of correlated information sources," *IEEE Trans. on Inform. Theory*, vol.IT-19, no. 4, pp.471-480, July 1973.
- [2] A. Aaron and B. Girod, "Compression with side information using turbo-codes," *In Proc. of Data Compress. Conf., DCC'02*, pp.252-261, April 2002.
- [3] J. Bajcsy and P. Mitran, "Coding for the Slepian-Wolf problem with turbo codes," *In Proc. of Global Telecomm. Conf., GLOBECOM'01*, vol.2, pp.1400-1404, Nov. 2001.
- [4] J. Bajcsy and I. Deslauriers, "Serial turbo coding for data compression and the Slepian-Wolf problem", *In Proc. of Inform. Theory Workshop*, pp.296-299, March 31-April 4 2003.
- [5] J. Garcia-Frias and Y. Zhao, "Compression of correlated binary sources using turbo-codes," *IEEE Communications Letters*, vol.5, no.10, pp.417-419, October 2001.
- [6] Y. Zhao and J. Garcia-Frias, "Joint estimation and compression of correlated nonbinary sources using punctured turbo codes," *IEEE Transactions on Communications*, vol.53, no.3, pp.385-390, March 2005.
- [7] A. D. Liveris, Z. Xiong, and C. N. Georgiades, "Distributed compression of binary sources using conventional parallel and serial concatenated convolutional codes," *In Proc. of IEEE Data Compression Conference*, 2003.
- [8] A. D. Liveris, Z. Xiong, and C. N. Georgiades, "Compression of binary sources with side information at the decoder using LDPC codes," *IEEE Communications Letters*, vol.6, no.10, pp.440-442, October 2002.
- [9] A. D. Liveris, Z. Xiong, and C. N. Georgiades, "Compression of binary sources with side information using low-density parity-check codes," *In Proc. of IEEE GLOBECOM 2002*, vol. 2, pp.1300-1304, 17-21 Nov. 2002.
- [10] J. Garcia-Frias, "Joint source-channel decoding of correlated sources over noisy channels," *In Proc. of DCC 2001*, pp.283-292, March 2001.
- [11] J. Garcia-Frias, W. Zhong, and Y. Zhao, "Iterative decoding schemes for source and joint source-channel coding of correlated sources," *In Proc. of Asilomar Conference on Signals, Systems, and Computers*, November 2002.
- [12] J. Garcia-Frias and Y. Zhao, "Near-Shannon/Slepian-Wolf performance for unknown correlated sources over AWGN channels," *IEEE Transactions on Communications*, vol.53, no.4, pp.555-559, April 2005.
- [13] F. Daneshgaran, M. Laddomada, and M. Mondin, "Iterative joint channel decoding of correlated sources employing serially concatenated convolutional codes," *IEEE Transactions on Information Theory*, vol.51, no.7, pp.2721-2731, July 2005.
- [14] F. Daneshgaran, M. Laddomada, and M. Mondin, "LDPC-based channel coding of correlated sources with iterative joint decoding," *IEEE Transactions on Communications*, Vol. 54, No. 4, pp.577-582, April 2006.
- [15] T.J. Richardson and R.L. Urbanke, "Efficient encoding of low-density parity-check codes," *IEEE Transactions on Information Theory*, vol.47, no.2, pp.638 - 656, Feb. 2001.
- [16] Xiao-Yu Hu, E. Eleftheriou, and D.M. Arnold, "Progressive edge-growth Tanner graphs," *IEEE Global Telecommunications Conference*, vol.2, pp.995-1001, 25-29 Nov. 2001.

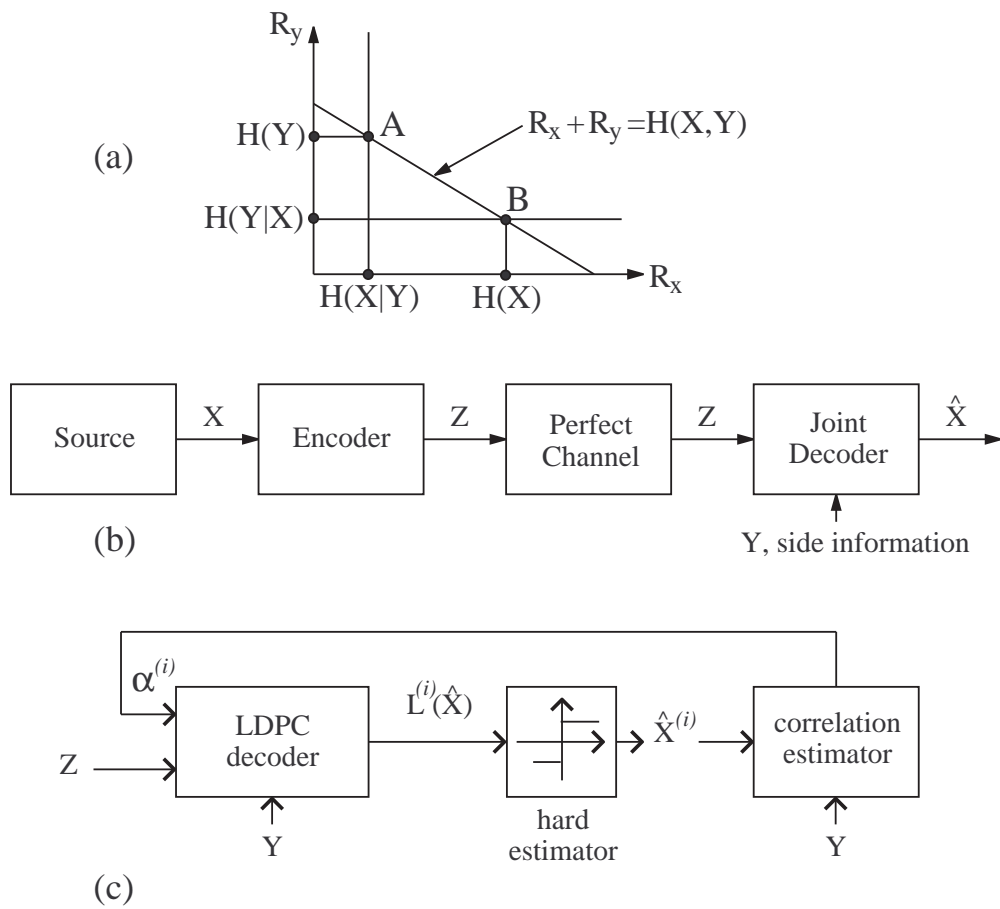


Fig. 1. Rate region for Slepian-Wolf encoding (a). Architecture of the encoder and joint decoder for the Slepian-Wolf problem (b). Architecture of the Iterative Joint decoder of correlated sources (c).

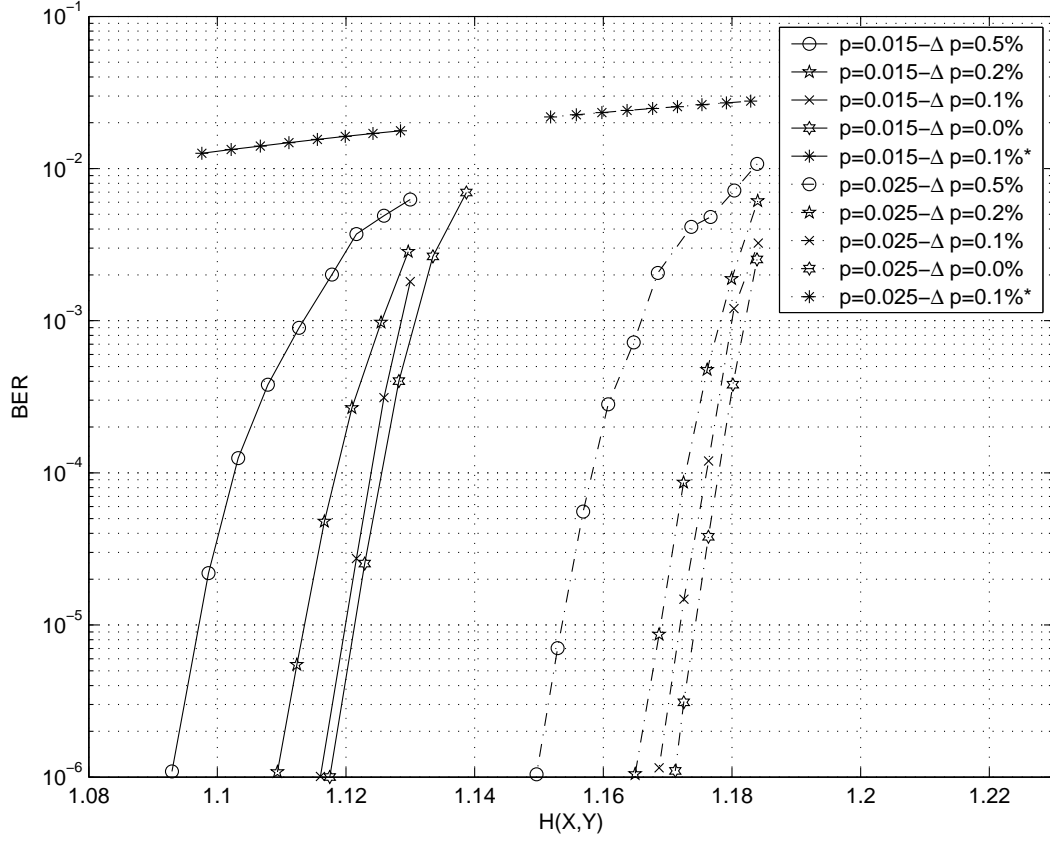


Fig. 2. BER performance of the proposed iterative decoding algorithm for a maximum of 5 global iterations as a function of the joint entropy between sources X and Y , when the stopping criterion for global iterations is applied. Results refer to the LDPCs labelled L_4 and L_3 in Table I. The legend shows the mean correlation value p and the maximum value of the correlation variation with respect to the mean value. Curves labelled with $*$ refer to the ones obtained without the iterative paradigm, using the mean correlation value.

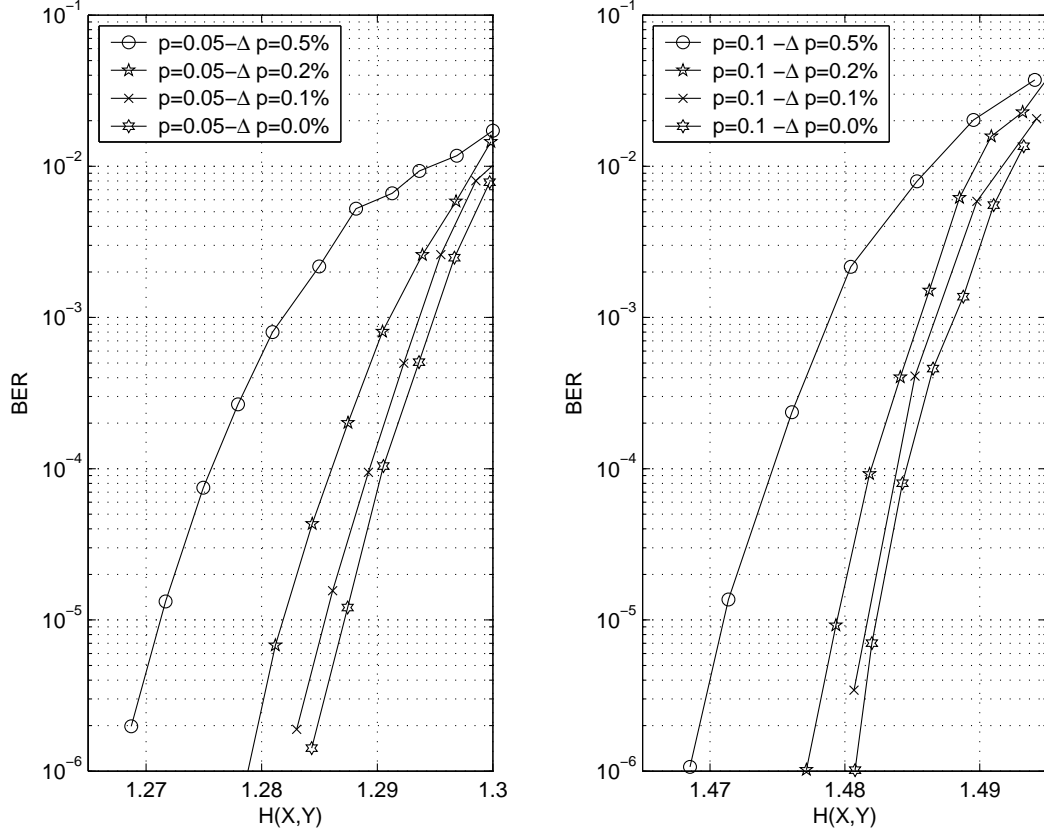


Fig. 3. BER performance of the proposed iterative decoding algorithm for a maximum of 5 global iterations as a function of the joint entropy between sources X and Y , when the stopping criterion for global iterations is applied. Results refer to the LDPCs labelled L_2 (left subplot) and L_1 (right subplot) in Table I. The legend shows the mean correlation value p and the maximum value of the correlation variation with respect to the mean value.

TABLE I
PARAMETERS OF THE DESIGNED LDPCs.

LDPC	k	n	R_X	d_v	d_c
L_1	16400	26200	0.597	3	8
L_2	16400	22400	0.365	3.21	12
L_3	16400	20300	0.237	3.45	18
L_4	16400	19500	0.189	3.0	19

TABLE II
COMPRESSION RATE PERFORMANCE OF THE ITERATIVE ALGORITHM FOR VARIOUS JOINT ENTROPIES.

p	0.015	0.025	0.05	0.1
$H(p) + 1$	1.112	1.169	1.286	1.469
R [5]	-	1.31	1.435	1.63
R [9]	-	1.276	1.402	1.60
$R = R_X + R_Y$	1.189 - L_4	1.237 - L_3	1.365- L_2	1.597- L_1